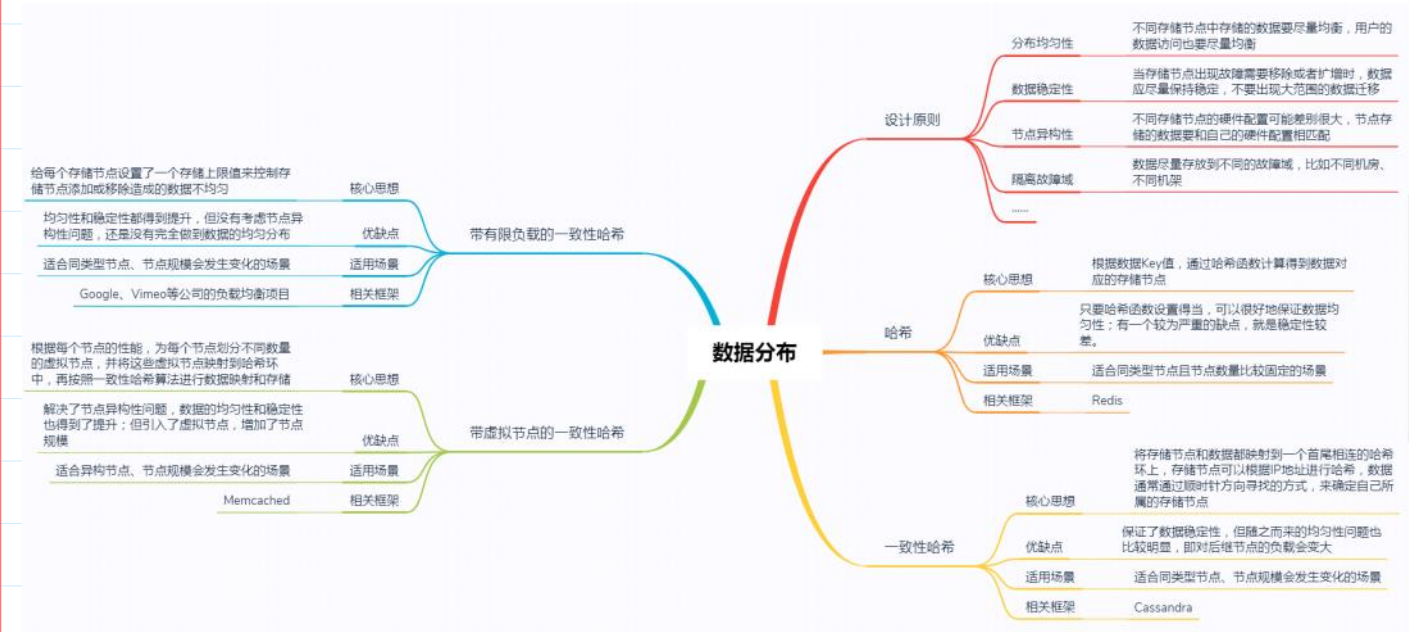


5.3 数据分布方式

2022年3月14日 15:59



数据均匀

- 不同存储节点中存储的数据要尽量均衡
- 用户访问也要做到均衡，避免出现某一个或某几个节点的访问量很大，但其他节点却无人问津的情况

数据稳定

当存储节点出现故障需要移除或者扩增时，数据按照分布规则得到的结果应该尽量保持稳定，不要出现大范围的数据迁移。

节点异构性

不同存储节点的硬件配置可能差别很大。

隔离故障域

一个好的数据分布算法，应该为每个数据映射一组存储节点

我们通常通过备份来实现数据的可靠性。但如果每个数据及它的备份，被分布到了同一块硬盘或节点上。

性能稳定性

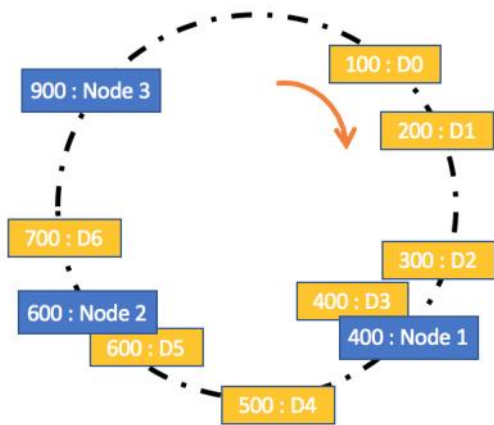
数据存储和查询的效率要有保证，不能因为节点的添加或者移除，造成存储或访问性能的严重下降。

数据分布方法

哈希

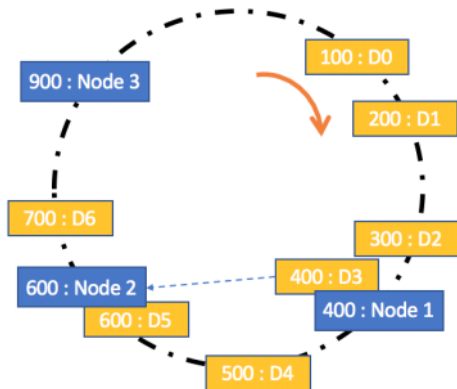
- 1. 优点：有好的hash函数可以保证很好数据均匀性
- 2. 缺点：稳定性差

一致性哈希



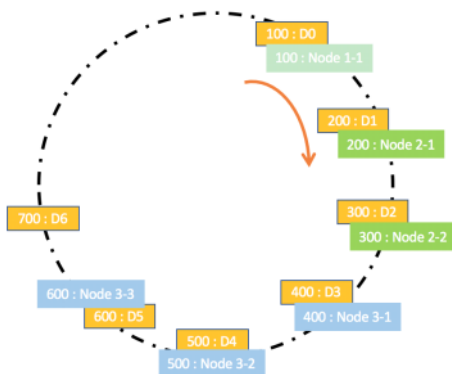
一致性哈希是对哈希方法的改进，在数据存储时采用哈希方式确定存储位置的基础上，又增加了一层哈希，也就是在数据存储前，对存储节点预先进行了哈希。

带有限负载的一致性哈希



给每个存储节点设置了一个存储上限值来控制存储节点添加或移除造成的数据不均匀。当数据按照一致性哈希算法找到相应的存储节点时，要先判断该存储节点是否达到了存储上限；如果已经达到了上限，则需要继续寻找该存储节点顺时针方向之后的节点进行存储。

带虚拟节点的一致性哈希



带虚拟节点的一致性哈希方法，核心思想是根据每个节点的性能，为每个节点划分不同数量的虚拟节点，并将这些虚拟节点映射到哈希环中，然后再按照一致性哈希算法进行数据映射和存储。

四种Hash对比

	均匀性	稳定性	节点异构性	典型应用
哈希算法	基本均匀	稳定性差	没有考虑节点异构	Redis集群
一致性哈希算法	均匀性较好，但可能会出现数据倾斜即某个节点承受成倍的压力	稳定性较好，节点的加入和退出不会造成数据大规模迁移	没有考虑节点异构	Cassandra集群
带有限负载的一致性哈希算法	均匀性较好，存储节点有上限，相比一致性哈希算法好一些	稳定性较好，节点的加入和退出不会造成大规模的数据迁移	没有考虑节点异构	Google Cloud Pub/Sub、HAProxy
带虚拟节点的一致性哈希算法	均匀性最好，每个存储节点拆分成多个虚拟节点，可以让数据分布更均匀	稳定性最好，每个存储节点拆分成多个虚拟节点，可以共同分担节点变化时对系统中原有节点的影响	考虑了节点异构性，性能好的存储节点可以对应更多的虚拟节点	Memcached缓存系统

数据分片和数据分区的区别

数据分区是从**数据存储块**的维度进行划分，不同的分区物理上归属于不同的节点。比如，现在有 2 个节点 Node1 和 Node2，2 个数据分区 Partition1 和 Partition2，Partition1 属于 Node1、Partition2 属于 Node2。

数据分片是从**数据**的维度进行划分，是指将一个数据集按照一定的方式划分为多个数据子集，不同的数据子集存在不同的存储块上，而这些存储块可以在不同的节点上，也可以在同一节点上。